NASA Contractor Report 181977
ICASE Report No. 90–3

ICASE

EULER AND NAVIER-STOKES COMPUTATIONS FOR TWO-DIMENSIONAL GEOMETRIES USING UNSTRUCTURED MESHES

D. J. Mavriplis

Contract No. NAS1–18605
January 1990

Institute for Computer Applications in Science and Engineering
NASA Langley Research Center
Hampton, Virginia 23665–5225

Operated by the Universities Space Research Association

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665-5225

EULER AND NAVIER-STOKES COMPUTATIONS FOR TWO-DIMENSIONAL GEOMETRIES USING UNSTRUCTURED MESHES

D. J. Mavriplis

Institute for Computer Applications in Science and Engineering
NASA Langley Research Center
Hampton, VA

ABSTRACT

A general purpose unstructured mesh solver for steady-state two-dimensional inviscid and viscous flows is described. The efficiency and accuracy of the method are enhanced by the simultaneous use of adaptive meshing and an unstructured multigrid technique. A method for generating highly stretched triangulations in regions of viscous flow is outlined, and a procedure for implementing an algebraic turbulence model on unstructured meshes is described. Results are shown for external and internal inviscid flows and for turbulent viscous flow over a multi-element airfoil configuration.

This research was supported under the National Aeronautics and Space Administration under NASA Contract Nos. NAS1-18107 and NAS1-18605 while the author was in residence at the Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center, Hampton, VA 23665.

1. INTRODUCTION

Numerical methods for the computation of steady-state compressible flows have progressed to the point where the major obstacle to achieving an efficient and accurate flow solution for a given problem lies in one's ability to generate an adequate mesh over the given geometry. For complex configurations, the most often employed approach consists of partitioning the domain into a number of topologically simple regions and generating a structured mesh in each of these regions. The construction of these so-called block-structured meshes has proven to be an expensive and time consuming process, requiring much human intervention.

An alternate approach is afforded by the use of unstructured meshes using triangular elements in two dimensions, and tetrahedral elements in three dimensions. Compared with structured meshes, in which grid lines must propagate across the entire domain, unstructured meshes are purely local constructions, and thus provide a much greater degree of flexibility in meshing complex geometries. Furthermore, they provide a natural setting for the use of adaptive meshing techniques, where new mesh points are added and the grid is restructured locally in regions where the flow gradients are large.

However, even in the case of inviscid flows, the use of unstructured meshes has often been impeded by sometimes questionable accuracy and inefficient solvers. The efficiency of these solvers is hindered by the use of indirect addressing required by the random data sets, and by the use of relatively simple solution algorithms, due to the difficulties associated in constructing implicit solvers on unstructured meshes, which require the inversion of large sparse matrices. While inefficiencies due to random data sets, which generally result in a reduction by a factor of three of the number of floating point operations per second (Mflop rate) achievable on present-day supercomputers, cannot be avoided, efficient solution algorithms requiring near optimal computational complexity may be devised. In the present work, this has been achieved through the use of a multigrid strategy [1]. Provided a consistent discretization of the governing equations is employed, unstructured meshes can be shown to provide extremely accurate solutions through extensive use of adaptive meshing [2,3,4,5].

For viscous flows, unstructured meshes have seldom been employed, or in certain cases, have been used only in the inviscid regions of the flow, as part of a hybrid approach where structured meshes are employed in the regions of viscous flow [6,7]. The strong directionality of the gradients in the viscous flow regions, and the requirements they impose on the mesh generation procedure, as well as the frequent use of algebraic turbulence models, appears to have provided a strong deterrent against the use of fully unstructured meshes for viscous flows.

In this paper, the development of an unstructured mesh solver for both inviscid and viscous steady-state flows about arbitrary two-dimensional geometries is described. This work represents an effort at constructing a general purpose, accurate, and efficient solution method. To this end, a general method for setting up the geometrical configuration (spline curves) and application of boundary conditions has been devised. An unstructured multigrid algorithm is used in conjunction with an adaptive meshing technique to ensure accurate and efficient solutions. For viscous flow calculations, a procedure for generating and adaptively refining highly stretched triangulations is presented. The implementation of an algebraic turbulence model for use on unstructured meshes is also described.

2. PROBLEM DEFINITION

The first step involves the definition of the problem to be solved, which relates to the definition of the geometry, and the specification of boundary conditions and initial conditions.

Geometry definition is the first step, which must necessarily be performed prior to the mesh generation and flow solution phases. In general, the geometrical configuration is described by an ordered series of points. These points are not employed as mesh points themselves. A spline curve which fits these points is constructed, and mesh points are taken at predetermined locations along this spline curve. Thus, the geometrical configuration is in fact defined by a series of spline curves. This is especially important when adaptive meshing techniques are considered. New points which are introduced on the boundary must conform to the spline definition of this boundary, rather than simply being positioned midway between two neighboring grid points. Hence the geometry definition stage consists of identifying ordered groups of geometry points which are to be splined, selecting a type of spline for each group, and generating the spline curve for each group, which is stored as a series of spline coordinates and coefficients, for later use in the mesh generation and flow solution phases.

The specification of boundary conditions forms part of the problem definition stage and as such, is best treated prior to the mesh generation and flow solution phases. The original edges which define the geometry (such edges link two geometry definition points prior to the spline fitting operation) are sorted into groups, each of which corresponds to a particular type of boundary condition. This information is then stored for subsequent use in the mesh generation and flow solution phase. When the initial mesh is generated, boundary points are assigned the boundary condition corresponding to that of the geometry definition edge from which they were generated. When adaptively adding points to an existing mesh, new boundary points may also be assigned a boundary condition type in this manner. In the flow solution phase, application of the boundary conditions consists of looping through all boundary points and applying the appropriate boundary conditions at each point, as dictated by the boundary condition type associated with each point. To efficiently vectorize this step, boundary points are sorted into groups, each group representing a specific type of boundary condition, and the appropriate boundary condition is then executed on each group in a vector fashion. This predetermination and storage of boundary condition types allows for a completely general flow solution algorithm and facilitates the adaptive insertion of new boundary points without upsetting the logic of the solver. Finally the specification of initial conditions, such as Mach number, Reynolds number, and angle of incidence, may be effected without loss of generality as an input list in the flow solution phase.

In this work, internal as well as external flow geometries have been considered. For external flow about multi-element airfoil configurations, each airfoil element is defined by a cubic spline. The circular far-field boundary is "splined" as a linear fit between a set of defining points. Tangential flow, or zero velocity boundary conditions are applied at the airfoil surfaces, depending on whether the Euler or Navier-Stokes equations are being solved. In the far field, a non-reflecting locally one-dimensional characteristic boundary condition is employed [8].

The internal flow about periodic cascade geometries requires the simultaneous use of two types of splines and three types of boundary conditions, thus providing a good illustration of the generality of this process. The initial definition of the geometry for this case, which is depicted in Figure 1, is composed of eight boundary regions. Regions 1 and 5 represent the inflow and outflow planes, regions 2,4 and 6,8 represent the periodic line. These regions are

thus all "splined" as straight line segments. Regions 3 and 7 represent the lower and upper surface of the turbine blade respectively, and are defined by cubic splines. In order to maintain continuity of slope and curvature at the leading and trailing edge points (i.e. the periodic points of the blade), in addition to the lower surface points, a periodic representation of the upper surface points is constructed and used in conjunction with these points to define the spline along the lower surface of the blade. A similar treatment is employed for the upper surface in boundary region 7. In regions 1 and 2, inlet and outlet boundary conditions are specified respectively. For the inlet flow, total pressure, total enthalpy, and the flow angle are specified, while the remaining condition is obtained by extrapolating the locally one-dimensional outgoing Riemann invariant normal to the boundary from the interior. For the outflow boundary, back pressure is specified, and total pressure, total enthalpy plus the outgoing locally one-dimensional Riemann invariant normal to the boundary are extrapolated from the interior [9,10]. In regions 3 and 7, flow tangency or zero velocity conditions are imposed, depending on whether inviscid or viscous flow solutions are sought. In regions 2, 4, 6, and 8, periodic boundary conditions are applied. A natural way of implementing periodic boundary conditions in the context of unstructured meshes is through the use of pointers. Each periodic boundary point is associated with its duplicate point on the corresponding periodic boundary through an integer pointer, which points to the address of the appropriate point. Thus, logically, the two corresponding periodic points refer to the same point, and are thus updated simultaneously and identically. However, duplicate physical copies of this point are required, each with a different physical coordinate, and each referring to a point on one of the two periodic cuts.

3. MESH GENERATION

3.1. Initial Mesh Generation

The initial unstructured mesh is generated in three essentially independent stages. In the first stage, a distribution of mesh points filling the domain is generated. These points are then joined together to form a set of non-overlapping triangles using a Delaunay triangulation algorithm. Finally, a post-processing operation is employed to smooth out the mesh by slightly repositioning the points according to an elliptic smoothing operator [11].

While adaptive meshing techniques can be relied upon to increase the mesh resolution in regions of high flow gradients, a good initial mesh point distribution is essential to ensure the capture of all salient flow features on the initial mesh, and to reduce the number of adaptivity cycles required to attain a given accuracy level. This is particularly true in the case of high Reynolds number viscous flows, where very small normal spacings are required in the viscous regions. Since much effort has been expended in devising structured mesh generation strategies for specific types of geometries, these provide a natural starting point for the generation of a mesh point distribution. Each component of the geometry may be fitted locally with a structured mesh, suitable to that particular type of component, and the union of all the points from these overlapping local meshes, which lie in the flow field, used as the basis for the triangulation. For multi-element airfoils, an O-mesh is fitted around each airfoil element using a hyperbolic mesh generation algorithm [12]. For internal flow cascade geometries, the initial mesh point distribution is derived from a structured H-mesh [10].

Delaunay triangulation represents a unique way of joining a set of points in a plane together to form a set of non-overlapping triangles. Bowyer's algorithm [13,14] is used to construct the triangulation. Assuming an initial triangulation exists (this may be constructed by

joining a small number of boundary points together), the mesh points are introduced and triangulated one at a time into the existing triangulation. Bowyer's algorithm makes use of the circumcircle property of a Delaunay construction, which states that the circumcircles of the triangles may not contain vertices from other triangles. Thus, each time a new mesh point is introduced, the union of all triangles whose circumcircles contain this new point is identified. The existing mesh structure is removed in this region and new triangles are formed by joining the new point to all the vertices of the restructured region. When all mesh points have been introduced and triangulated, the initial unstructured mesh is obtained.

3.2. Adaptive Mesh Generation

This sequential insertion and local restructuring of new mesh points is ideally suited for an adaptive meshing strategy. Once a solution has been obtained on the initial mesh, new mesh points are created midway along mesh edges in regions of large flow gradients. Each new mesh point is then triangulated into the existing mesh using Bowyer's algorithm. The search for the triangles whose circumcircles are intersected by the new point can be made extremely efficient by beginning with the two triangles on either side of the edge within which the new point was generated, and then searching through neighboring triangles. Hence, adaptive mesh enrichment may be accomplished using only local searching and restructuring, thus avoiding the need for global mesh regeneration. When new boundary points are created, they must be positioned on the spline curve which defines that boundary. For concave boundaries, this results in mesh points which are not enclosed by any of the existing mesh triangles, as shown in Figure 2. To avoid failure of the intersected circumcircle search routine, new boundary points are initially positioned midway along the boundary edge within which they are generated. The circumcircle search and local retriangulation are then effected. The new mesh point is then displaced onto the spline curve, thus forming a sliver triangle which joins the new point with the two ends of the generating mesh edge. Such sliver triangles, which lead to a crossing of grid lines for concave boundaries (c.f. Figure 2), and for convex boundaries represent elements exterior to the computational domain, must be identified and subsequently removed.

4. FLOW SOLUTION

In conservative form, the full Navier-Stokes equations read

$$\frac{\partial w}{\partial t} + \frac{\partial f_c}{\partial x} + \frac{\partial g_c}{\partial y} = \frac{\sqrt{\gamma} M_\infty}{\mathrm{Re}_\infty} \left[\frac{\partial f_v}{\partial x} + \frac{\partial g_v}{\partial y} \right] \tag{1}$$

where w is the solution vector and f_c and g_c are the cartesian components of the convective fluxes

$$w = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{bmatrix} \quad f_c = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uE + up \end{bmatrix} \quad g_c = \begin{bmatrix} \rho v \\ \rho vu \\ \rho v^2 + p \\ \rho vE + vp \end{bmatrix} \tag{2}$$

In the above equations, ρ represents the fluid density, u and v the x and y components of fluid velocity, E the total energy, and p is the pressure which can be calculated from the equation of state of a perfect gas

$$p = (\gamma - 1)\rho \left[E - \frac{(u^2+v^2)}{2} \right] \tag{3}$$

The viscous fluxes f_v and g_v are given by

$$f_v = \begin{bmatrix} 0 \\ \sigma_{xx} \\ \sigma_{xy} \\ u\sigma_{xx}+v\sigma_{xy}-q_x \end{bmatrix} \qquad g_v = \begin{bmatrix} 0 \\ \sigma_{xy} \\ \sigma_{yy} \\ u\sigma_{yx}+v\sigma_{yy}-q_y \end{bmatrix} \tag{4}$$

where σ represents the stress tensor, and q the heat flux vector, which are given by the constitutive equations for a Newtonian fluid

$$\begin{aligned} \sigma_{xx} &= 2\mu u_x - \frac{2}{3}\mu(u_x+v_y) \\ \sigma_{yy} &= 2\mu v_y - \frac{2}{3}\mu(u_x+v_y) \\ \sigma_{xy} &= \sigma_{yx} = \mu(u_y+v_x) \\ q_x &= -k\frac{\partial T}{\partial x} = -\frac{\gamma}{\gamma-1}\frac{\mu}{Pr}\frac{\partial \frac{p}{\rho}}{\partial x} \\ q_y &= -k\frac{\partial T}{\partial y} = -\frac{\gamma}{\gamma-1}\frac{\mu}{Pr}\frac{\partial \frac{p}{\rho}}{\partial y} \end{aligned} \tag{5}$$

γ is the ratio of specific heats of the fluid, M_∞ the freestream Mach number, Re_∞ the Reynolds number based on the airfoil chord, and Pr the Prandtl number. The coefficient of viscosity μ varies with the temperature of the fluid, and is calculated as

$$\mu = K\,T^{0.72} \tag{6}$$

where K is a constant. Equation (1) represents a set of partial differential equations which must be discretized in space in order to obtain a set of coupled ordinary differential equations, which can then be integrated in time to obtain the steady-state solution.

The spatial discretization procedure begins by storing flow variables at the vertices of the triangles. The stress tensor σ and the heat flux vector q must be calculated at the centers of the triangles. This is achieved by computing the required first differences in the flow variables (from equations (5)) at the triangle centers. For a piecewise linear approximation of the flow variables in space, the first differences are constant over each triangle, and may be computed as

$$w_x = \frac{1}{A}\iint \frac{\partial w}{\partial x}\,dxdy = \frac{1}{A}\int w\,dy = \frac{1}{A}\sum_{k=1}^{3}\frac{w_{k+1}+w_k}{2}(y_{k+1}-y_k) \tag{7}$$

$$w_y = \frac{1}{A}\iint \frac{\partial w}{\partial y}\,dxdy = \frac{1}{A}\int w\,dx = \frac{1}{A}\sum_{k=1}^{3}\frac{w_{k+1}+w_k}{2}(x_{k+1}-x_k) \tag{8}$$

where the summation over k refers to the three vertices of the triangle. The flux balance equations are obtained by a Galerkin finite-element type formulation. The Navier-Stokes equations are first rewritten in vector notation

$$\frac{\partial w}{\partial t} + \nabla.\mathbf{F}_c = \frac{\sqrt{\gamma}M_\infty}{\mathrm{Re}_\infty}\,\nabla.\mathbf{F}_v \tag{9}$$

where the bold typeset denotes vector quantities. F_c is a dyadic (second order tensor), the

cartesian components of which are given by the f_c and g_c convective flux vectors defined previously, and a similar notation is employed for the viscous flux terms. Multiplying by a test function ϕ, and integrating over physical space yields

$$\frac{\partial}{\partial t}\iint_\Omega \phi w \, dxdy + \iint_\Omega \phi \nabla.\mathbf{F}_c \, dxdy = \frac{\sqrt{\gamma}M_\infty}{\mathrm{Re}_\infty} \iint_\Omega \phi \nabla.\mathbf{F}_v \, dxdy \tag{10}$$

Integrating the flux integrals by parts, and neglecting boundary terms gives

$$\frac{\partial}{\partial t}\iint_\Omega \phi w \, dxdy = \iint_\Omega \mathbf{F}_c .\nabla\phi \, dxdy - \frac{\sqrt{\gamma}M_\infty}{\mathrm{Re}_\infty} \iint_\Omega \mathbf{F}_v .\nabla\phi \, dxdy \tag{11}$$

In order to evaluate the flux balance equations at a vertex P, ϕ is taken as a piecewise linear function which has the value 1 at node P, and vanishes at all other vertices. Therefore, the integrals in the above equation are non-zero only over triangles which contain the vertex P, thus defining the domain of influence of node P, as shown in Figure 3. To evaluate the above integrals, we make use of the fact that ϕ_x and ϕ_y are constant over a triangle, and may be evaluated as per equations (7) and (8). The convective fluxes $\mathbf{F}_c$ are taken as piecewise linear functions in space, and the viscous fluxes $\mathbf{F}_v$ are piecewise constant over each triangle, since they are formed from first derivatives in the flow variables. Evaluating the flux integrals with these assumptions, one obtains

$$\frac{\partial}{\partial t}\iint_\Omega \phi w \, dxdy = \sum_{e=1}^{n} \frac{\mathbf{F}_c^A + \mathbf{F}_c^B}{6}.\Delta\mathbf{L}_{AB} - \frac{\sqrt{\gamma}M_\infty}{\mathrm{Re}_\infty} \sum_{e=1}^{n} \frac{\mathbf{F}_v^e}{2}.\Delta\mathbf{L}_{AB} \tag{12}$$

where the summations are over all triangles in the domain of influence, as shown in Figure 3. Δ_{AB} represents the directed (normal) edge length of the face of each triangle on the outer boundary of the domain, $\mathbf{F}_c^A$ $\mathbf{F}_c^B$ are the convective fluxes at the two vertices at either end of this edge, and $\mathbf{F}_v^e$ is the viscous flux in triangle e, e being a triangle in the domain of influence of ϕ. If the integral on the left hand side of equation (12) is evaluated in the same manner, the time derivatives become coupled in space. Since we are not interested in the time-accuracy of the scheme, but only in the final steady-state solution, we employ the concept of a lumped mass matrix. This is equivalent to assuming w to be constant over the domain of influence while integrating the left hand side. Hence, we obtain

$$\Omega_p \frac{\partial w_p}{\partial t} = \sum_{e=1}^{n} \frac{\mathbf{F}_c^A + \mathbf{F}_c^B}{2}.\Delta\mathbf{L}_{AB} - \frac{\sqrt{\gamma}M_\infty}{\mathrm{Re}_\infty} \sum_{e=1}^{n} \frac{3}{2} (\mathbf{F}_v^e .\Delta\mathbf{L}_{AB}) \tag{13}$$

where the factor of 1/3 is introduced by the integration of ϕ over the domain, and Ω_p represents the surface area of the domain of influence of P. For the convective fluxes, this procedure is equivalent to the vertex finite-volume formulation described in [1,11]. For a smoothly varying regular triangulation, the above formulation is second-order accurate.

Additional artificial dissipation terms are required to ensure stability and to capture shocks without producing numerical oscillations. This is necessary for both inviscid and viscous flow computations, since in the later case, large regions of the flow field behave essentially inviscidly and the physical viscosity is not sufficient to guarantee numerical stability for the type of mesh spacings typically employed. Artificial dissipation terms are thus constructed as a blend of a Laplacian and a biharmonic operator in the conserved flow variables. The Laplacian term represents a strong formally first-order accurate dissipation which is turned on only in the vicinity of a shock, and the biharmonic term represents a weaker second-order accurate dissipation which is employed in regions of smooth flow [11,15]. The spatially

discretized equations are integrated in time to obtain the steady-state solution using a five-stage time-stepping scheme, where the convective terms are evaluated at each stage within a time step, and the dissipative terms (both physical and artificial) are only evaluated at the first, third, and fifth stages. This particular scheme has been designed to maintain stability in regions where the flow is dominated by viscous effects, and to rapidly dampen out high-frequency error components, which is an essential feature for a scheme intended to drive a multigrid algorithm [16,17]. Convergence is accelerated by making use of local time-stepping, implicit residual averaging [16], and an unstructured multigrid algorithm [1].

The idea of a multigrid strategy is to accelerate the convergence to steady-state of a fine grid solution through corrections computed on coarser grids. An initial time step is performed on the fine grid, and the flow variables and residuals are then transferred to the coarse grid. A correction equation is constructed on the coarse grid by adding a forcing function to the original discretized equations. This forcing function is formed by taking the difference between the transferred residuals and the residuals of the transferred variables, thus ensuring that the evolution of the coarse grid equations is driven by the fine grid residuals. Hence, when the fine grid residuals vanish, the coarse grid equations are identically satisfied, and generate zero corrections. After transferring values down from the fine grid, a time step is performed on the coarse grid, and the new values are transferred down to the next coarser grid. When the coarsest grid is reached, the computed corrections are successively interpolated back up to the finest grid, and the entire cycle is repeated. In the context of unstructured meshes, a sequence of coarse and fine meshes is best constructed by generating the individual meshes independently from one another (as opposed to subdividing a coarse mesh). Thus, in general, the coarse and fine meshes of a given sequence do not have any common mesh points or nested elements. Thus, the patterns for transferring the variables, residuals, and corrections back and forth between the various meshes of the sequence must be determined in a preprocessing operation, where an efficient tree-search algorithm is employed [1].

Such a multigrid algorithm may be combined with an adaptive meshing strategy in a natural manner. First, a sequence of globally generated meshes is constructed, and multigrid time-stepping is performed on this sequence until a satisfactorily converged solution is obtained. At this point, a new adaptively refined mesh is generated, and the transfer patterns for transferring variables from the previous mesh to the new mesh are determined. The flow variables are then transferred to this new mesh, providing a starting solution, and multigrid time-stepping is resumed on this new sequence which now contains an additional fine mesh. The process may be repeated, as shown in Figure 4, each time adding a new finer mesh to the sequence, until a converged solution of the desired accuracy is obtained.

5. INVISCID FLOW RESULTS

When the viscous terms in the Navier-Stokes equations are neglected, the Euler equations are obtained. Inviscid flow calculations can thus be computed using the method previously described, but neglecting the terms on the right-hand-side of equations (1) and/or (9), and replacing the no-slip wall boundary condition with a tangential slip velocity boundary condition. Inviscid flow computations can be performed for a significantly lower cost than viscous flow computations, not only due to the reduced number of terms which need to be discretized, but also due to the elimination of the boundary layer and wake regions, where extremely high gradients generally occur, and which must be resolved in the viscous case.

5.1. External Flow Geometry

The first test case consists of the computation of the inviscid compressible flow about a high-lift three-element airfoil configuration. The free-stream Mach number is 0.2, and the incidence is 8 degrees. At these conditions, the flow is entirely subcritical. However, an extreme double suction peak occurs at the leading edge of the main airfoil, where the flow becomes nearly sonic. The capture of this extremely high localized gradient requires a very fine mesh resolution in this region. A sequence of seven meshes were used in the multigrid algorithm. The first three meshes were generated globally, and the further four meshes were generated by adaptive refinement. The criterion for adaptive mesh enrichment is based on the undivided difference of density [18]. The first difference of density computed along each mesh edge is examined. When this difference is larger than some fraction of the RMS average of all density differences across the mesh, a new point is added midway along the edge. A second pass is then performed which splits the remaining edges of each triangular element bordering on a previously split edge, thus ensuring an isotropic refinement. The finest mesh of this calculation is depicted in Figure 5. It contains 11949 nodes, of which 512 are on the airfoil surfaces. Extreme refinement is seen to occur in the main airfoil leading edge region and in the gap regions. A globally refined mesh of this resolution would have required 10 to 20 times more points, and thus would be prohibitively expensive. The coarsest mesh of the sequence contains merely 110 points. The computed Mach contours in the flow field are depicted in Figure 6, where the high gradients in the leading-edge region are evident. In Figure 7, a comparison of the computed surface pressure distribution with that generated by a finite-element full potential solver [19], shows good agreement between the two methods, and illustrates the magnitude of the suction peak, where the pressure coefficient rapidly attains a value of -16.0. The convergence history for this case is depicted in Figure 8, where the fine grid residuals were reduced down to machine zero (in double precision) in just over 300 multigrid cycles. Each multigrid cycle required roughly 1.8 CPU seconds on a single processor of a Cray-2 supercomputer, so that engineering calculations (50 - 100 mg cycles) could be obtained in 2 to 3 CPU minutes.

5.2. Internal Flow Calculations

In a second test case, the inviscid flow through a turbine blade cascade geometry has been computed. The particular blade geometry has been the subject of an experimental and computational investigation at the occasion of a VKI lecture series [20]. A total of seven meshes were used in the multigrid algorithm, with the last three meshes generated adaptively, using the undivided density difference criterion. The coarsest mesh of the sequence contains only 51 points, while the finest mesh, depicted in Figure 9, contains 9362 points. Extensive mesh refinement can be seen to occur in the neighborhood of shocks, and in other regions of high gradients. The inlet flow incidence is 30 degrees, and the average inlet Mach number is 0.27. The flow is turned 96 degrees by the blades, and the average exit isentropic Mach number is 1.3. At these conditions, the flow becomes supersonic as it passes through the cascade, and a complex oblique shock wave pattern is formed. These are evident from the computed Mach contours depicted in Figure 10. All shocks are well resolved, including some of the weaker reflected shocks, which non-adapted mesh computations often have difficulty resolving [10]. Details of the flow in the rounded trailing edge region of the blade are shown in Figure 10, where oblique shock waves are formed on the upper and lower surfaces of the blade, and where the flow separates (inviscidly), forming a small recirculation region. The

surface isentropic Mach number distribution for this case is compared with experimental data provided from [20], in Figure 11. The upper surface experimental values correspond to an exit isentropic Mach number of 1.31, while the lower surface experimental values are taken from a case where the measured exit isentropic Mach number was 1.21. These values are included since lower surface data was not available for an exit Mach number of 1.31, and since the lower surface values were seen to be relatively insensitive to the exit Mach number in this range. Keeping in mind the inviscid nature of the computational results, good agreement is observed over most of the surface of the blade. The irregularities in the numerical solution are due to a poor surface definition of the blade, the effect of which is enhanced by the adaptive meshing procedure, which tends to refine the mesh in the vicinity of non-smooth geometries. Once the first four globally generated meshes were constructed, the entire flow solution - adaptive mesh enrichment cycle was performed three times, executing 25 multigrid cycles at each stage. This entire operation required 40 CPU seconds on a single processor of a Cray-YMP supercomputer. The residuals on the finest mesh were reduced by two and a half orders of magnitude, which should be adequate for engineering calculations. The efficiency of this solution illustrates the possibility of constructing a truly interactive adaptive mesh Euler solver in two-dimensions, provided the present algorithm may be efficiently implemented in parallel on all eight processors of the Cray-YMP.

6. CONSIDERATIONS FOR HIGH REYNOLDS NUMBER VISCOUS FLOWS

While the methodology previously described applies in principle to viscous flows as well, the efficient solution of high Reynolds number flows requires the generation of highly stretched meshes as well as the implementation of a turbulence model for use on unstructured meshes.

6.1. Mesh Generation

The generation of highly stretched unstructured meshes requires a suitable mesh point distribution, with closely packed points in the normal direction, and sparsely distributed points in the streamwise direction, as well as a method for producing an appropriate triangulation of such a point distribution. The generation of a suitable point distribution may be effected as previously described, using local hyperbolically generated structured meshes. However, a Delaunay triangulation of a given set of points tends to produce the most equiangular triangles possible, and therefore in general, is not well suited for the generation of highly stretched mesh elements. Thus, an alternate triangulation procedure must be employed. The approach taken consists of defining a stretching vector (stretching magnitude and direction) at each node of the initial point distribution throughout the flow field. Assuming an initial triangulation has been obtained, when a new mesh point is to be inserted, the associated stretching vector is employed to construct a locally mapped space such that, within this mapped space, the local point distribution appears isotropic. A Delaunay triangulation is then performed to triangulate the new point into the mesh in this mapped space, and the resulting triangulation is mapped back into physical space, thus resulting in the desired stretched triangulation [21]. Hence, a fully unstructured mesh with highly stretched elements in the boundary layer and wake regions, nearly equilateral triangles in the inviscid regions of flow, and a smooth variation of elements throughout the transition regions is obtained. The use of fully unstructured meshes for viscous flow calculations has been pursued, as opposed to the hybrid structured-unstructured meshes often advocated in the literature [6,7], due to the increased generality they afford in dealing with geometries with close tolerances between neighboring bodies, where confluent boundary

layers may occur, and due to the ease with which adaptive meshing may be incorporated throughout the viscous and inviscid regions of flow.

6.2. Turbulence Modeling for Unstructured Meshes

Algebraic turbulence models typically require information concerning the distance of each mesh point from the nearest wall. Turbulence length scales, which are related to the local boundary layer or wake thickness, are determined by scanning the appropriate flow values along specified streamwise stations. For example, the Baldwin-Lomax turbulence model [22] uses the location of the maximum of the moment of vorticity along streamwise stations normal to the boundary layer to estimate the local turbulence length scales. In the context of unstructured meshes, mesh points and thus flow variables do not naturally occur at regular streamwise locations. Thus, lines normal to the walls and viscous layers must be created and flow variables interpolated onto these lines, in order that the turbulence length scales may be determined. This type of approach has previously been implemented for supersonic ramp geometries by Rostand [23]. However, in the present work, more complex geometries must be accommodated. Recalling that, in the mesh generation procedure, the initial mesh point distribution was obtained by generating a series of local structured meshes about each geometry component, a natural manner of creating streamwise turbulence modeling stations is to make use of these local structured meshes as background meshes for the turbulence modeling routine. Thus, at each time step in the solution procedure, the current flow variables from the global unstructured mesh are interpolated onto each of the local structured meshes. The Baldwin-Lomax turbulence model is executed on these local structured meshes, and the resulting eddy viscosity distribution is interpolated back onto the unstructured mesh. In regions where two or more local structured meshes overlap, the multiple eddy viscosity values (one from each local mesh) which are interpolated back to the unstructured mesh are each weighted by their relative distance from the respective wall. Structured mesh lines emanating from one geometry component are terminated if they intersect a neighboring component, as shown in Figure 12, such that in any region of the flow field, the eddy viscosity is only related to the viscous layers and associated walls which are directly visible from that point. The same interpolation routines employed for the multigrid algorithm are used to pass variables back and forth between the global unstructured mesh and the local structured turbulence meshes. The determination of these transfer patterns is done in an efficient preprocessing operation, and the transfer addresses and coefficients are stored for subsequent use in the turbulence modeling routine. The whole process is very efficient, and in general, the entire turbulence modeling routine, including the interpolation procedures requires only 15% of the total time within a multigrid cycle. Memory requirements are however increased by about 50%, since extra variables and transfer coefficients must be stored for the local structured meshes.

7. TURBULENT FLOW RESULTS

The above modifications have been incorporated into the present scheme in order to compute the turbulent flow in the transonic regime past a two-element airfoil. The configuration consists of a main airfoil with a leading edge slat, which has been the subject of extensive wind tunnel tests [24], as part of a program aimed at improving the maneuvering capabilities of fighter aircraft in the transonic regime. A multigrid sequence of five meshes was employed to compute the flow about this configuration. The finest mesh is depicted in Figure 13. It contains 22,509 points, of which 256 lie on the surface of the main airfoil, and 128 on the surface

of the slat. The average width of the elements on the airfoil surfaces is 0.00001 chords, resulting in cell aspect ratios of the order of 1000:1 in these regions. The turbulence background meshes, consisting of one local structured mesh for each airfoil component, are depicted in Figure 12. The computed Mach contours are shown in Figure 14. For this case, the freestream Mach number is 0.5, the Reynolds number based on the chord is 4.5 million, and the incidence is 2.8 degrees. At these conditions, the flow is supercritical, and a shock is formed on the upper surface of the slat, as can be seen from the figure. The computed shock is somewhat diffuse, and an increased mesh resolution in this region would be required to obtain a crisper shock definition. However, the sudden thickening of the boundary layer as it interacts with the shock is evident from the computed Mach contours. A small recirculation region is also observed on the lower surface of the slat. A comparison of the computed surface pressure distribution with the experimental wind tunnel data is given in Figure 15. Computed and experimental values are seen to agree favorably in all regions, demonstrating a good prediction of the suction peaks, and location of the slat upper surface shock. This solution required roughly eight cpu minutes on a single processor of a Cray-2 supercomputer, which corresponds to 75 multigrid cycles on the finest grid, during which the residuals were reduced by approximately three orders of magnitude. To the author's knowledge, this represents the first compressible turbulent flow calculation for multi-element airfoil geometries using unstructured meshes.

8. CONCLUSION

A method for solving viscous and inviscid flows about arbitrary two-dimensional configurations has been presented. An attempt has been made to keep the method as general as possible. The simultaneous use of multigrid and adaptive meshing results in a rapidly convergent and accurate solution. For a given number of unknowns (mesh points), unstructured mesh solutions can be obtained with roughly the same number of operations as is required by the most efficient current structured mesh solvers. However, the speed of execution of unstructured mesh codes on present-day vector computers is roughly three times slower than that observed with structured mesh codes, due to the indirect addressing and scatter-gather operations required by the use of random data-sets. However, this factor can easily be outweighed through the use of a more efficient placement of grid points, using adaptive meshing techniques. For inviscid computations, the present algorithm appears robust and efficient enough that it may be implemented in an interactive mode on the latest generation of supercomputers. An inviscid solver based on these techniques in three dimensions, which is in the planning stages, should also provide a competitive solution technique for large problems. For turbulent viscous flow calculations, substantial modifications to the mesh generation phase were required. The implementation of an algebraic turbulence model has demonstrated a good prediction capability for flow over streamlined bodies. In future work, the implementation of a more general turbulence model, such as a field equation model, will be considered for flow over arbitrary geometries with massive separation.

9. ACKNOWLEDGMENTS

This work was made possible in large part due to the computational facilities made available by the National Aerodynamic Simulation (NAS) facility, and support provided by the Theoretical Aerodynamics Branch (TAB) of NASA Langley.

REFERENCES

1. Mavriplis, D. J., "Multigrid Solution of the Two-Dimensional Euler Equations on Unstructured Triangular Meshes", *AIAA Journal*, Vol 26, No. 7, July 1988, pp. 824-831

2. Mavriplis, D. J., "Accurate Multigrid Solution of the Euler Equations on Unstructured and Adaptive Meshes", *AIAA paper 88-3707, First National Fluid Dynamics Congress*, Cincinnati, Ohio, July 24-28, 1988.

3. Lohner, R., Morgan, K., Peraire, J., Vahdati, M., "Finite Element Flux-Corrected Transport for the Euler and Navier-Stokes Equations", *Int. J. Num. Meth. Fluids* Vol 7, 1987, pp. 1093-1109

4. Holmes, D. G., and Lamson, H. S., "Adaptive Triangular Meshes for Compressible Flow Solutions", *Proc. of the First International Conference on Numerical Grid Generation in Computational Fluid Dynamics*, Eds. J. Hauser and C. Taylor, Pineridge Press, 1986.

5. Stoufflet, B., Periaux, J., Fezoui, F., and Dervieux, A., "Numerical Simulation of 3-D Hypersonic Euler Flows Around Space Vehicles Using Adapted Finite Elements", *AIAA paper 87-0560* January 1987.

6. Holmes D. G., and Connell, S., "Solution of the 2-D Navier-Stokes Equations on Unstructured Adaptive Grids" *AIAA paper 89-1932, Proc. of the AIAA 9th Computational Fluid Dynamics Conference, Buffalo, NY*, June, 1989.

7. Nakahashi, N., "FDM-FEM Zonal Approach for Viscous Flow Computations Over Multiple Bodies", *AIAA paper 87-0604*, January, 1987.

8. Thomas, J. L., and Salas, M. D, "Far-Field Boundary Conditions for Transonic Lifting Solutions to the Euler Equations", *AIAA paper 85-0020* January 1985.

9. Jameson, A. and Liu, F., "Multigrid Calculations for Cascades", *Lecture Notes in Physics, Proc. of the 11th International Conference on Numerical Methods in Fluid Dynamics, Williamsburgh, VA, 1988*, Eds. D. Dwoyer, Y. Hussaini, R. Voigt, (Springer Verlag, 1988), pp. 318-325.

10. Arnone, A. and Swanson, R. C., "A Navier-Stokes Solver for Cascade Flows", *ICASE Rep. 99-32, NASA CR 181682*, July, 1988.

11. Mavriplis, D. J., "Solution of the Two-Dimensional Euler Equations on Unstructured Triangular Meshes" *Ph.D Thesis*, Department of Mechanical and Aerospace Engineering, Princeton University, June 1987.

12. Cordova, J. Q., and Barth, T. J., "Grid Generation for General 2-D Regions Using Hyperbolic Equations", *AIAA paper 88-0520*, January, 1988.

13. Bowyer, A., "Computing Dirichlet Tessalations", *The Computer Journal*, Vol. 24, No. 2, 1981, pp. 162-166

14. Baker, T. J., "Three Dimensional Mesh Generation by Triangulation of Arbitrary Point Sets", *Proc. of the AIAA 8th Comp. Fluid Dyn. Conf., AIAA paper 87-1124*, June, 1987

15. Jameson, A., Baker, T. J., and Weatherill, N. P., "Calculation of Inviscid Transonic Flow over a Complete Aircraft", *AIAA paper 86-0103*, January, 1986.

16. Mavriplis, D. J., Jameson, A., and Martinelli L., "Multigrid Solution of the Navier-Stokes Equations on Triangular Meshes", *AIAA paper 89-0120*, January, 1989.

17. Martinelli, L., "Calculations of Viscous Flows with a Multigrid Method", *Ph.D Thesis*, Department of Mechanical and Aerospace Engineering, Princeton University, October, 1987.

18. Dannenhoffer, J. F., and Baron, J. R., "Grid Adaptation for the 2-D Euler Equations", *AIAA paper 85-0484*, January, 1985.

19. Wigton, L. B., *Private Communication*, The Boeing Company

20. Sieverding, C. H., "Experimental Data on Two Transonic Turbine Blade Sections and Comparisons with Various Theoretical Methods", *Transonic Flows in Turbomachinery*, VKI Lecture Series 59, 1973.

21. Mavriplis, D. J., "Adaptive Mesh Generation for Viscous Flows Using Delaunay Triangulation" *ICASE Rep. 88-47, NASA CR 181699*, To appear in Journal of Comp. Physics, 1990.

22. Baldwin, B. S., Lomax, H., "Thin Layer Approximation and Algebraic Model for Separated Turbulent Flows", *AIAA paper 78-275*, 1978

23. Rostand, P. "Algebraic Turbulence Models for the Computation of Two-Dimensional High Speed Flows Using Unstructured Grids", *ICASE Rep 88-63*, Submitted to Numerical Methods in Fluids, November 1988.

24. Volpe, G., "A Multigrid Method for Computing the Transonic Flow Over Two Closely-Coupled Airfoil Components", *ICAS paper 84-1.4.3* Paper presented at the 14th ICAS Congress, Toulouse, France, September, 1984.

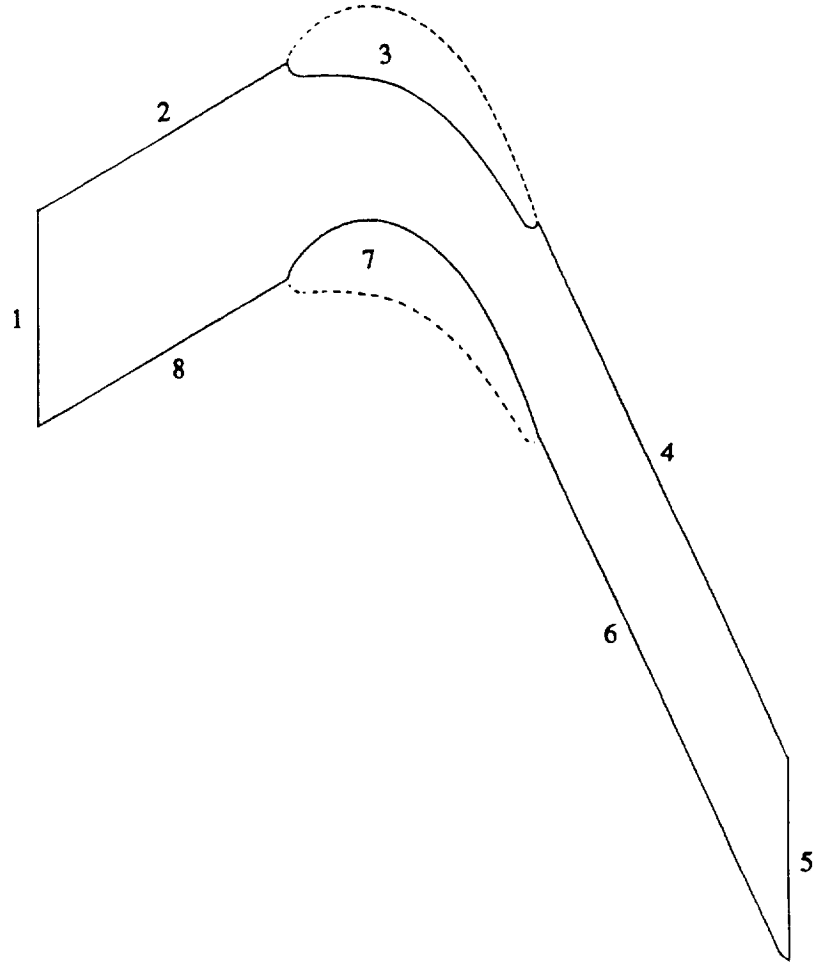

Figure 1
Definition of the Various Boundary Regions for the Periodic Turbine Blade Cascade Geometry

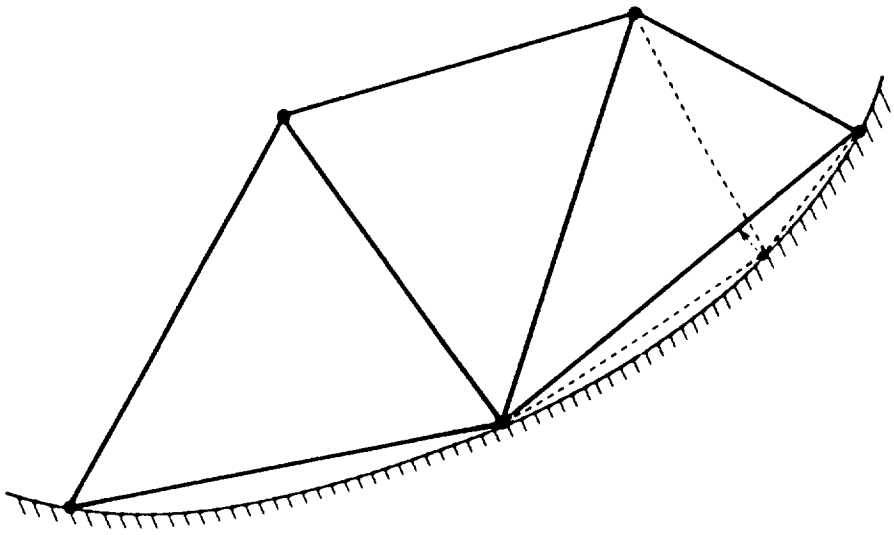

Figure 2
Illustration of an Adaptively Inserted Boundary Point in the
Region of a Concave Boundary and Resulting Sliver Cross-Over Triangle

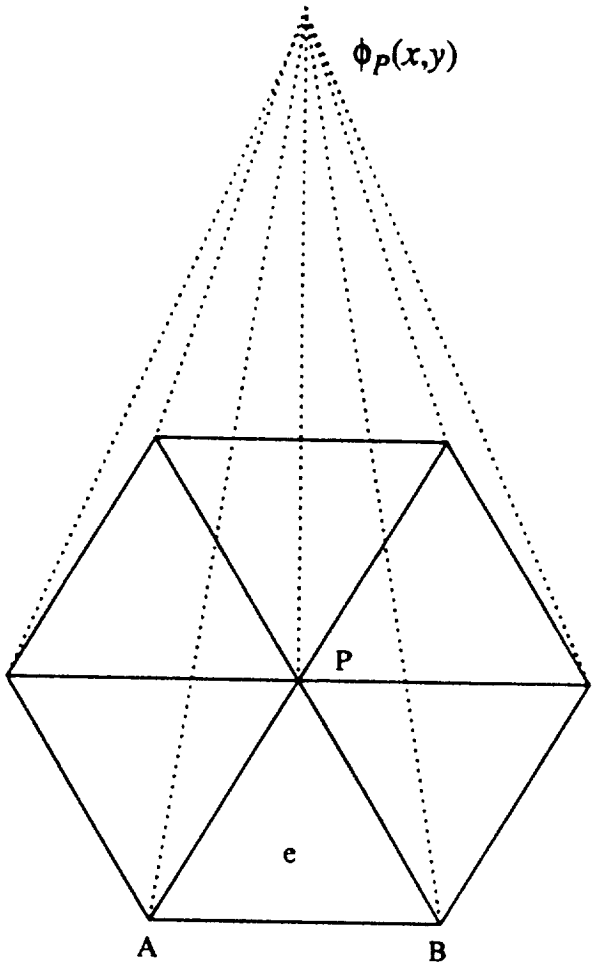

Figure 3
Domain of Influence of Finite-Element Basis Function and Equivalent Finite-Volume Control Volume

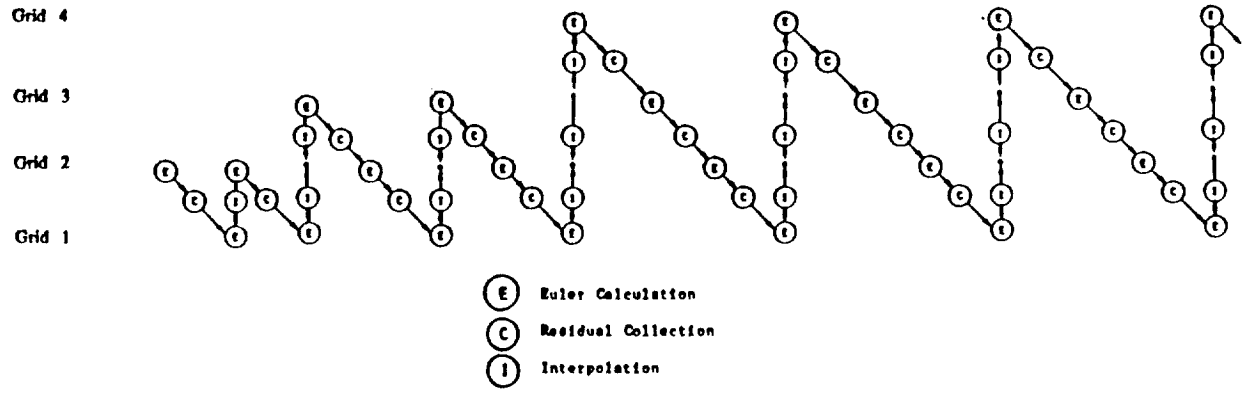

Figure 4
Full Multigrid Algorithm Employed in Conjunction with Adaptive Meshing Strategy

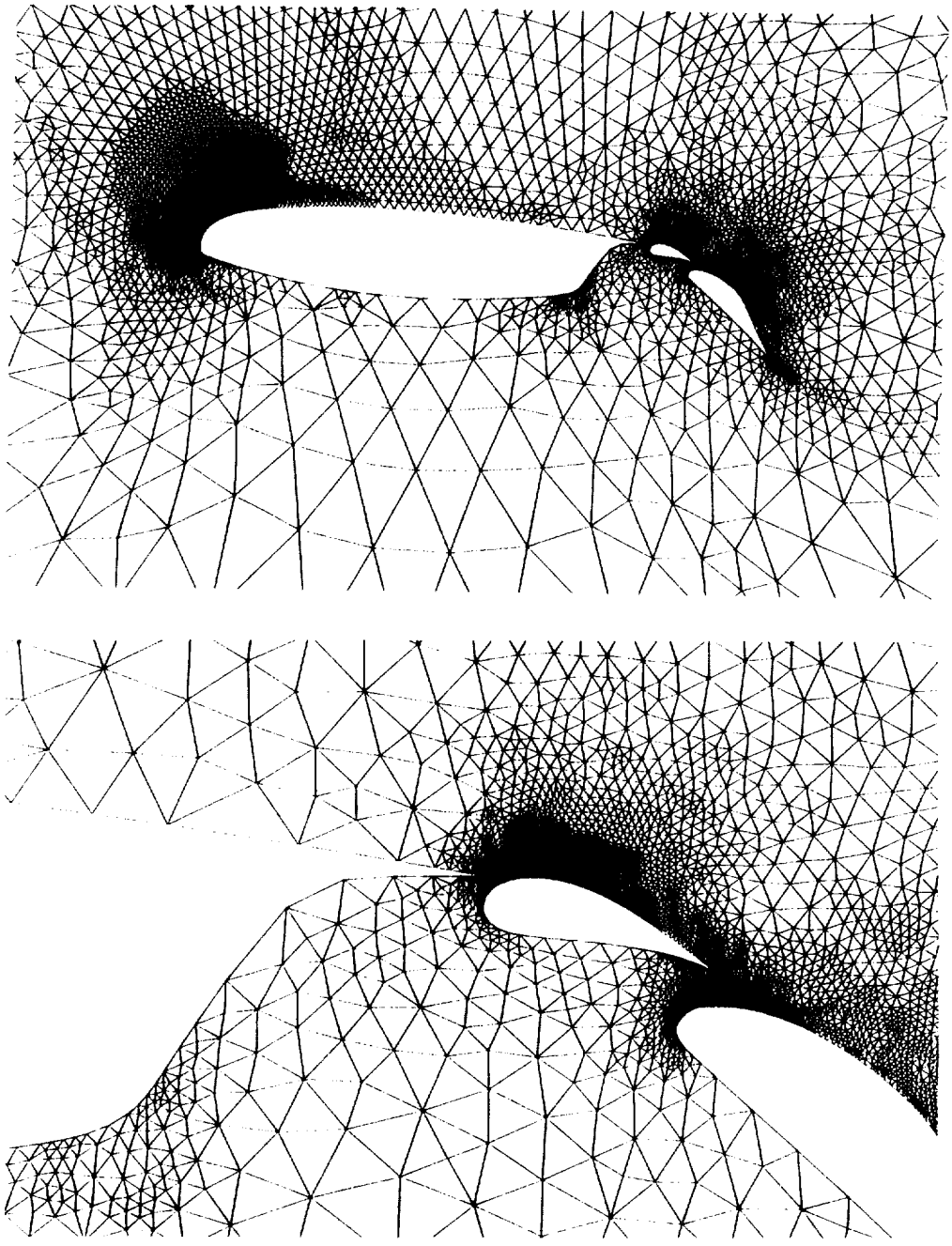

Figure 5
Adapted Mesh Employed for Computing Inviscid Subcritical Flow over a Three-Element Airfoil Configuration; Number of Nodes = 11949

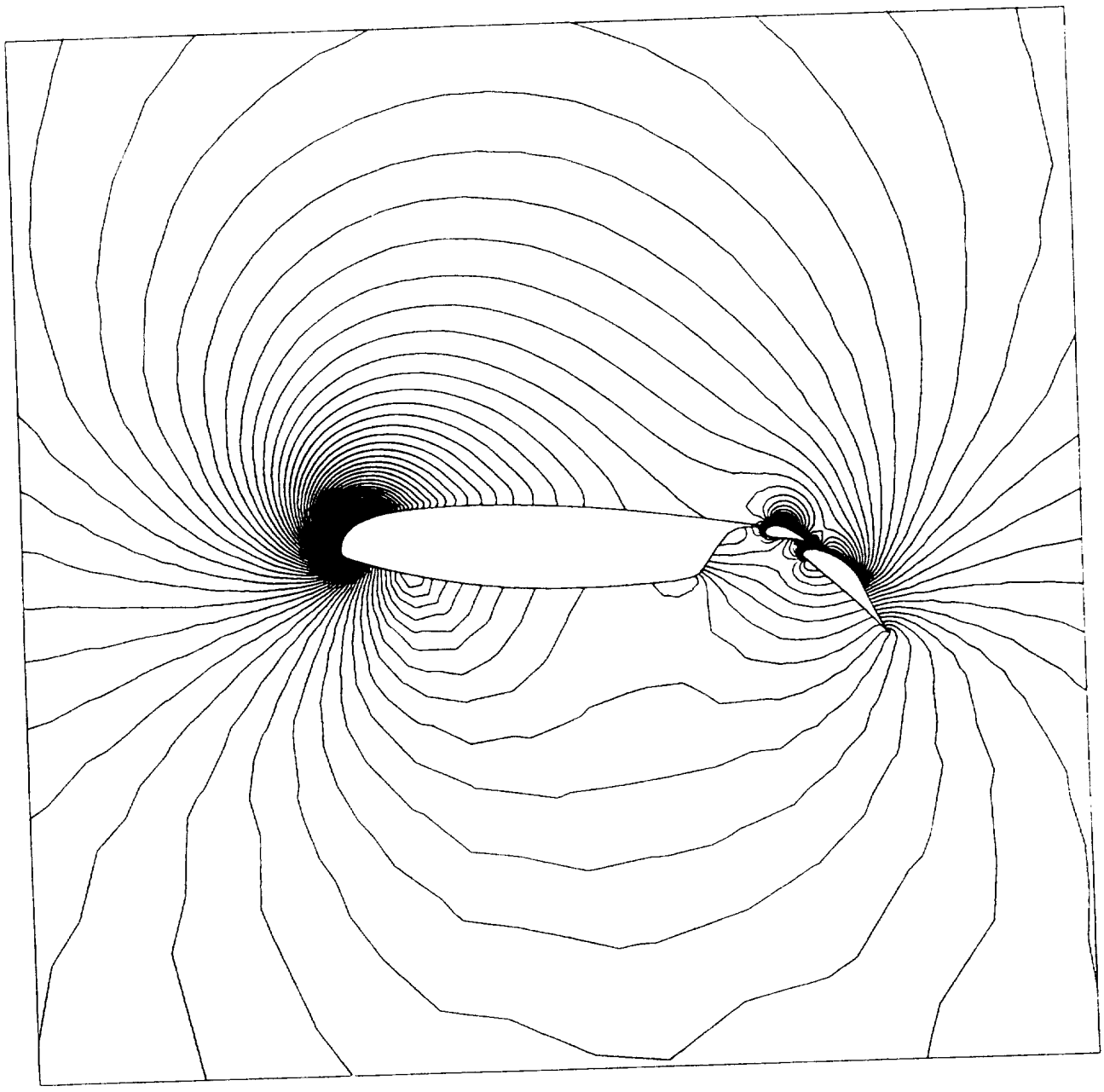

Figure 6
Computed Mach Contours for Inviscid Subcritical Flow over a Three Element Airfoil Configuration
Mach = 0.2, Incidence = 8 degrees

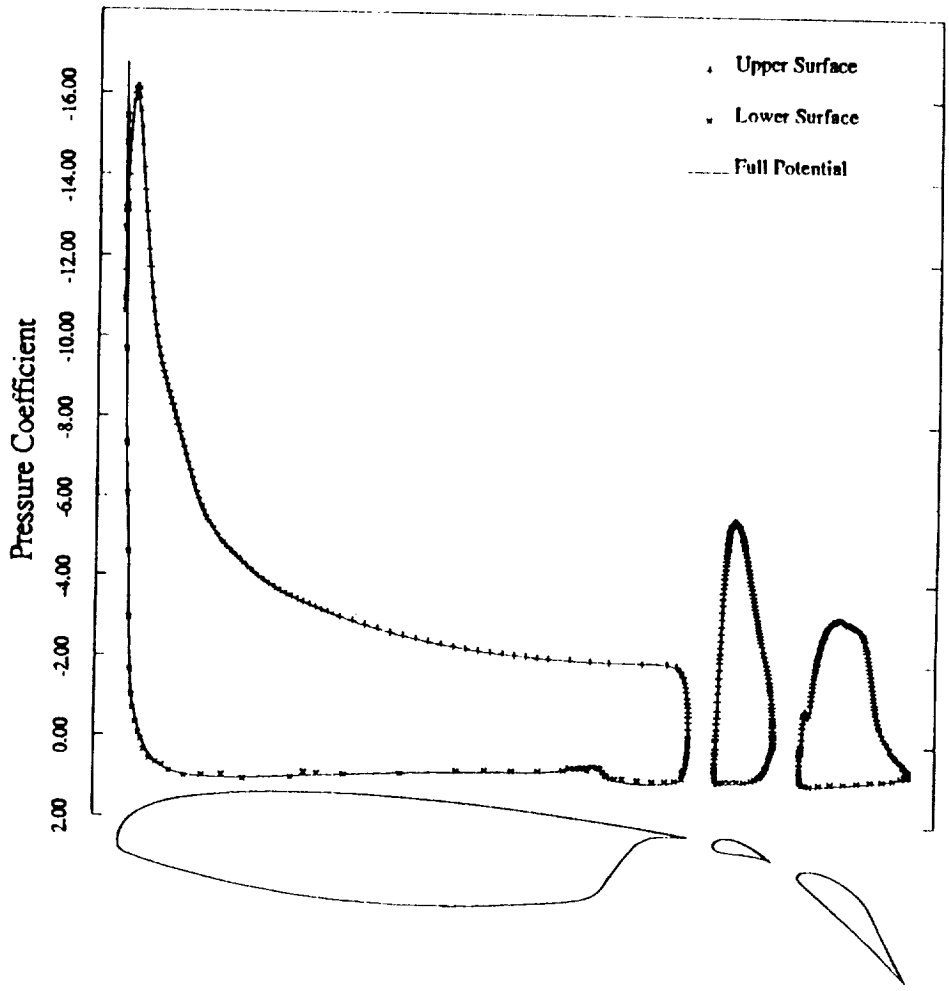

Figure 7
Comparison of the Computed Surface Pressure Distribution for the Present Euler Solution with that of a Full Potential Solution from [19]; Mach = 0.2, Incidence = 8 degrees

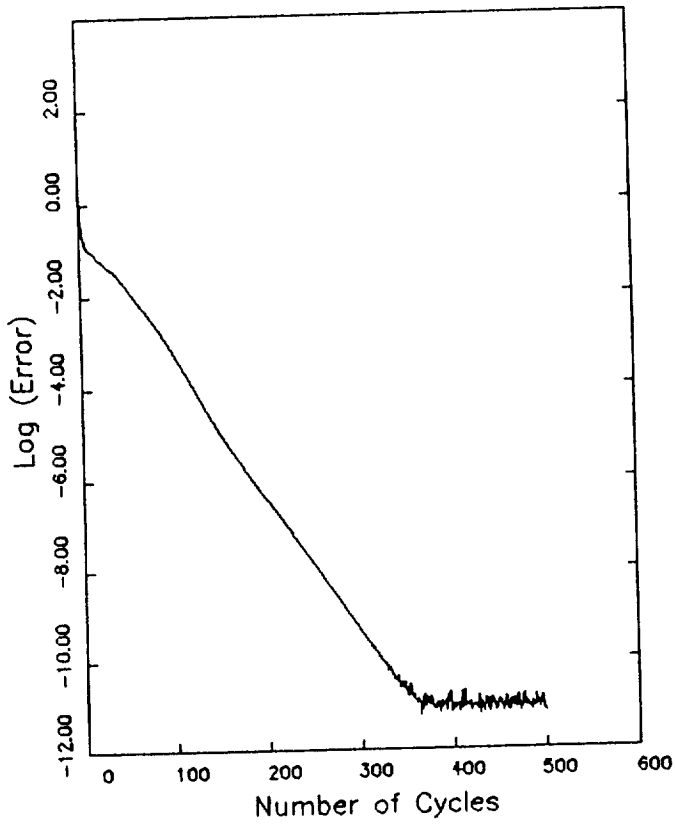

Figure 8
Convergence Rate on Finest Mesh for the Three-Element Airfoil Case
as Measured by the Average of the Density Residuals throughout the Flowfield

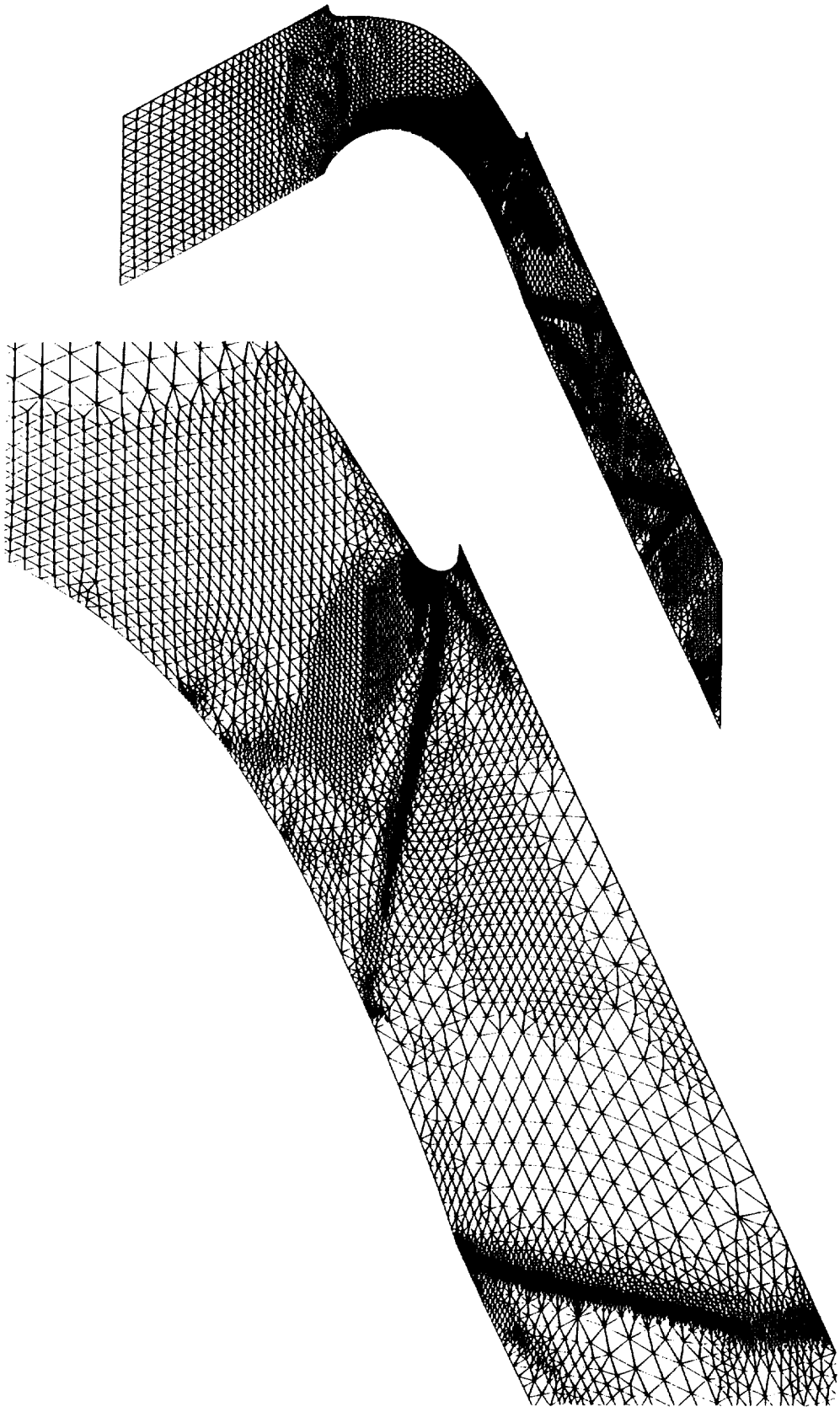

Figure 9
Adaptive Mesh Employed for Computing Transonic Inviscid Flow Through a Periodic Turbine Blade Cascade Geometry; Number of Nodes = 9362

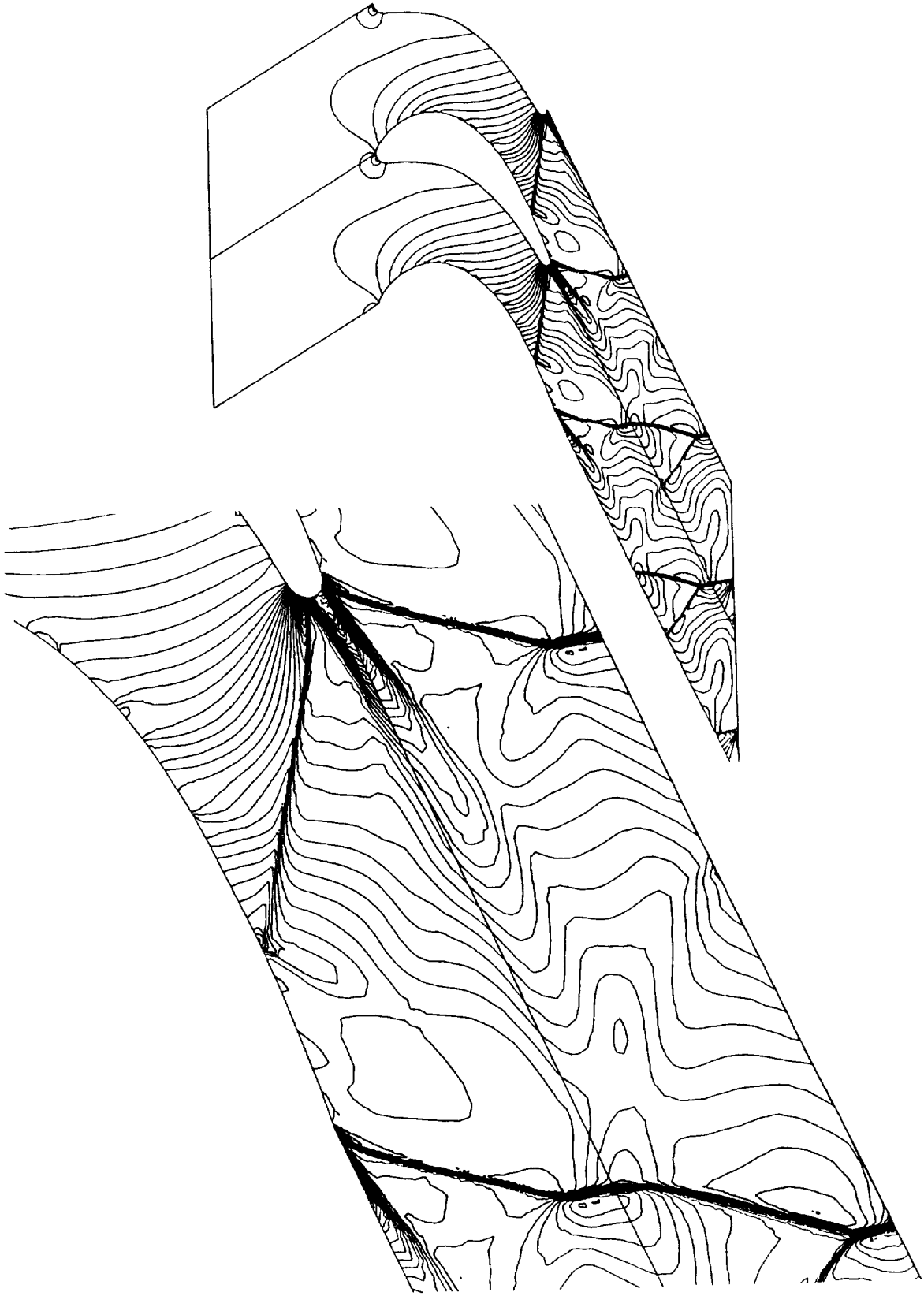

Figure 10
Computed Mach Contours for Flow Through a Periodic Turbine Blade Cascade Geometry

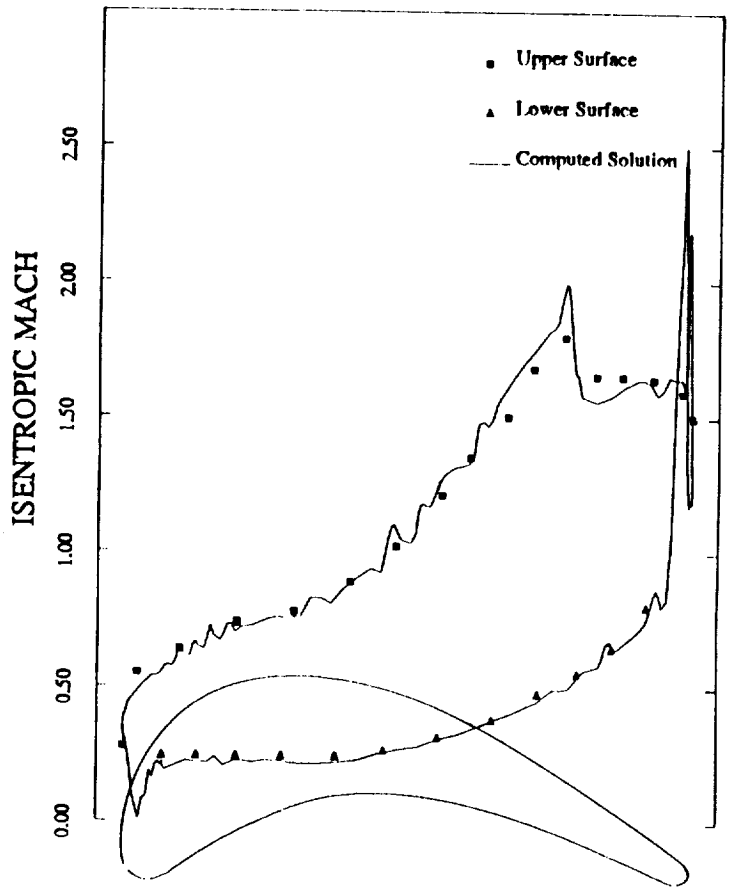

Figure 11
Comparison of Computed Surface Isentropic Mach Number Distribution with Experimental Values for Flow Through a Periodic Turbine Blade Cascade Geometry

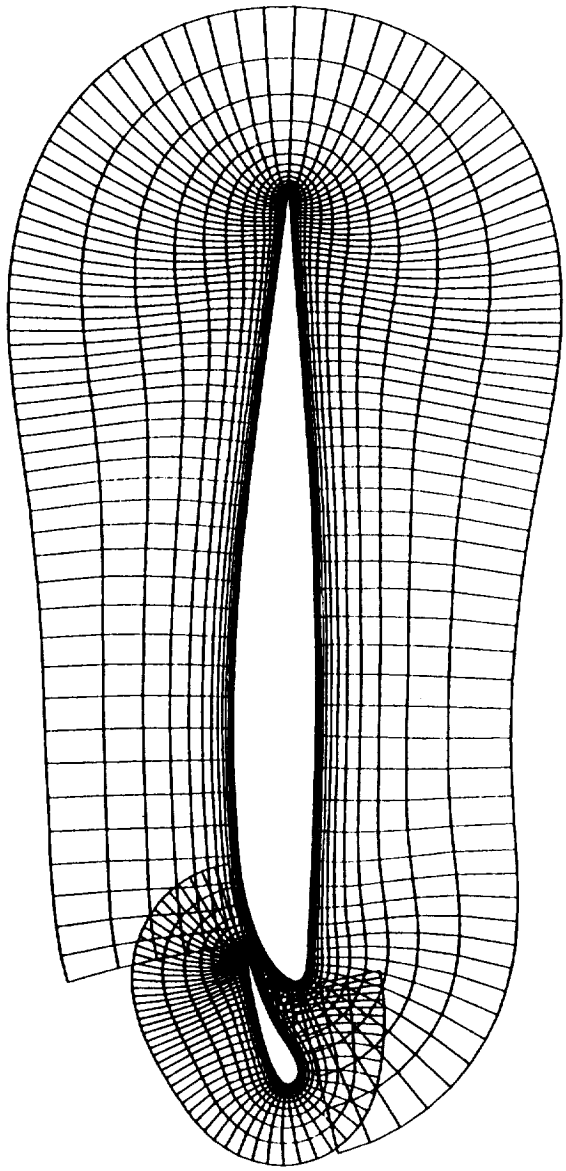

Figure 12
Illustration of Local Structured Background Turbulence Modeling Meshes Employed for Computing Turbulent Flow over a Two-Element Airfoil Configuration

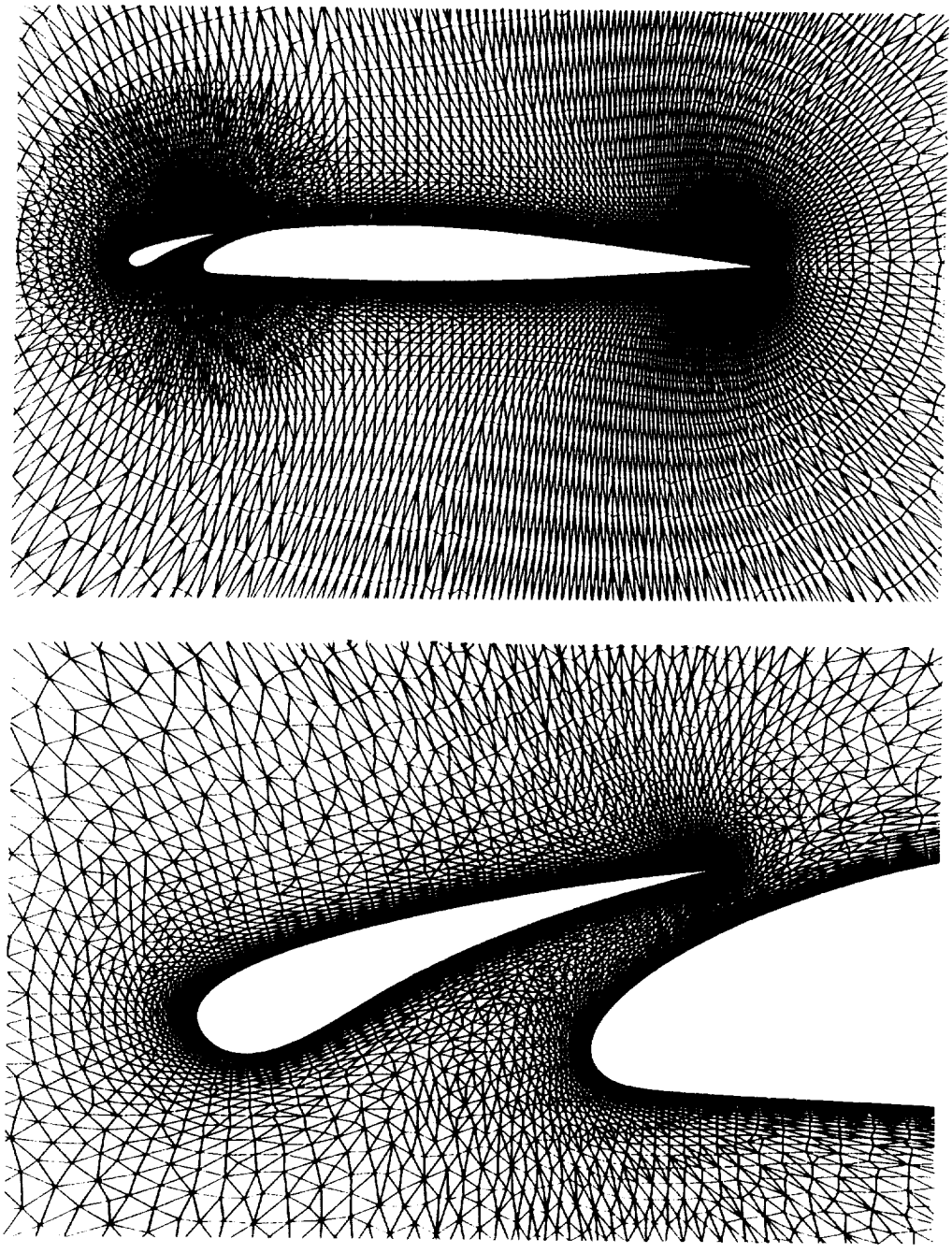

Figure 13
Fully Unstructured Mesh Employed for Computing Supercritical Turbulent Viscous Flow over a Two-Element Airfoil Configuration; Number of Nodes = 22,509

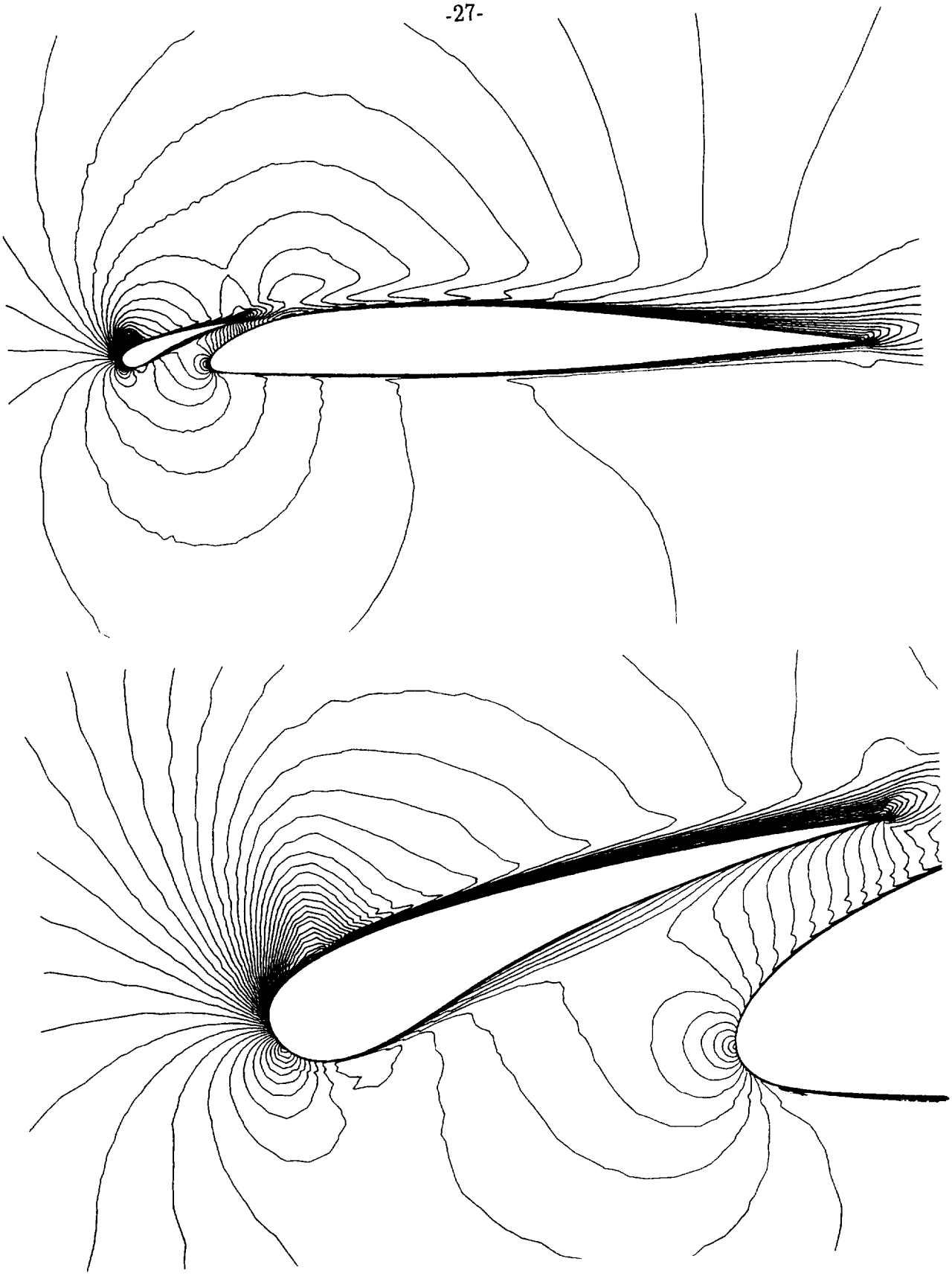

Figure 14
Computed Mach Contours of Supercritical Turbulent Viscous Flow over a Two-Element Airfoil Configuration
Mach Number = 0.5, Reynolds Number = 4.5 million, Incidence = 2.8 degrees

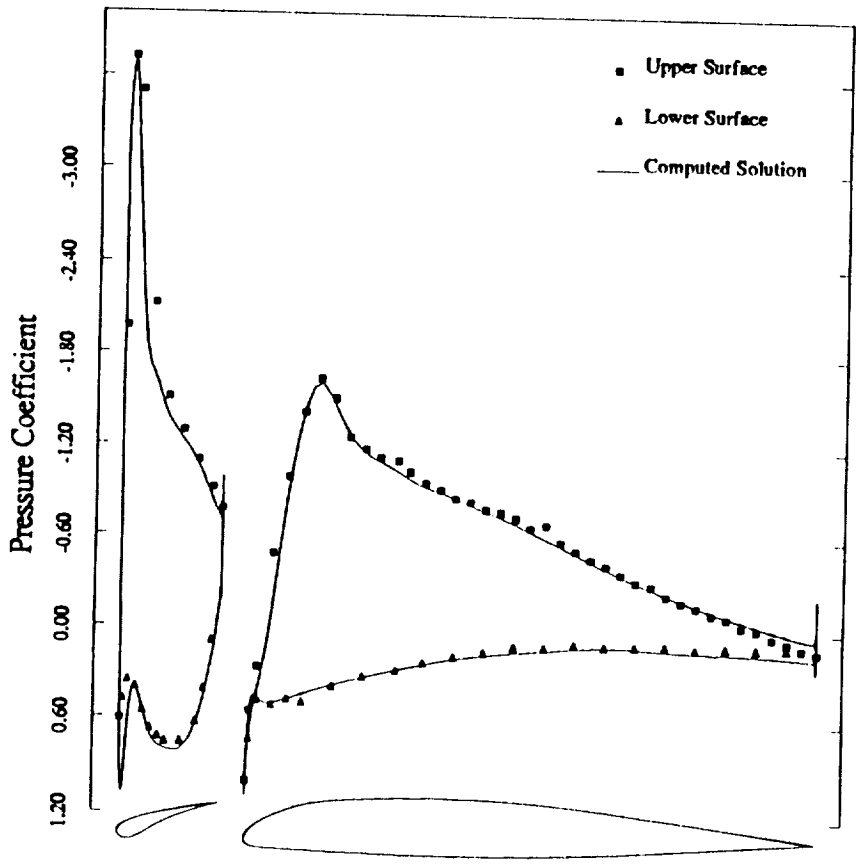

Figure 15
Comparison of Computed Surface Pressure Distribution with Experimental Wind-Tunnel Data for Flow Over Two-Element Airfoil Configuration
Mach Number = 0.5, Reynolds Number = 4.5 million, Incidence = 2.8 degrees

NASA National Aeronautics and Space Administration

Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CR-181977 ICASE Report No. 90-3		

4. Title and Subtitle	5. Report Date
EULER AND NAVIER-STOKES COMPUTATIONS FOR TWO-DIMENSIONAL GEOMETRIES USING UNSTRUCTURED MESHES	January 1990
	6. Performing Organization Code
7. Author(s)	8. Performing Organization Report No.
D. J. Mavriplis	90-3
	10. Work Unit No. 505-90-21-01
9. Performing Organization Name and Address Institute for Computer Applications in Science and Engineering Mail Stop 132C, NASA Langley Research Center Hampton, VA 23665-5225	11. Contract or Grant No. NAS1-18605
	13. Type of Report and Period Covered Contractor Report
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Langley Research Center Hampton, VA 23665-5225	14. Sponsoring Agency Code

15. Supplementary Notes

Langley Technical Monitor:
Richard W. Barnwell

Final Report

Submitted to the Canadian Aeronautics

16. Abstract

A general purpose unstructured mesh solver for steady-state two-dimensional inviscid and viscous flows is described. The efficiency and accuracy of the method are enhanced by the simultanious use of adaptive meshing and an unstructured multigrid technique. A method for generating highly stretched triangulations in regions of viscous flow is outlined, and a procedure for implementing an algebraic turbulence model on unstructured meshes is described. Results are shown for external and internal inviscid flows and for turbulent viscous flow over a multi-element airfoil configuration.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Euler, Navier-Stokes, Unstuctured	02 - Aeronautics 64 - Numerical Analysis Unclassified - Unlimited

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price
Unclassified	Unclassified	30	A03

NASA FORM 1626 OCT 86